

Developing Serious Games for Cultural Heritage: A State-of-the-Art Review

Eike Falk Anderson · Leigh McLoughlin ·
Fotis Liarokapis · Christopher Peters ·
Panagiotis Petridis · Sara de Freitas

Abstract Although the widespread use of gaming for leisure purposes has been well documented, the use of games to support cultural heritage purposes, such as historical teaching and learning, or for enhancing museum visits, has been less well considered. The state-of-the-art in serious game technology is identical to that of the state-of-the-art in entertainment games technology. As a result the field of serious heritage games concerns itself with recent advances in computer games, real-time computer graphics, virtual and augmented reality and artificial intelligence. On the other hand, the main strengths of serious gaming applications may be generalised as being in the areas of communication, visual expression of information, collaboration mechanisms, interactivity and entertainment. In this report, we will focus on the state-of-the-art with respect to the theories, methods and technologies used in serious heritage games. We provide an overview of existing literature of relevance to the domain, discuss the strengths and weaknesses of the described methods and point out unsolved problems and challenges. In addition, several case studies illustrating the application of methods and technologies used in cultural heritage are presented.

CR Subject Classification H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – Artificial, augmented, and virtual realities · I.2.1 [Artificial Intelligence]: Applications and Expert Systems – Games · K.3.1 [Computers and Education]: Computer Uses in Education – Computer-assisted instruction · K.8.0 [Personal Computing]: General – Games

1 Introduction

Computer games with complex virtual worlds for entertainment are enjoying widespread use and in recent years we have witnessed the introduction of serious games, including

E.F. Anderson & F. Liarokapis & C. Peters
Interactive Worlds Applied Research Group (iWARG), Coventry University, UK

L. McLoughlin
The National Centre for Computer Animation (NCCA), Bournemouth University, UK

P. Petridis & S. de Freitas
Serious Games Institute (SGI), Coventry University, UK

the use of games to support cultural heritage purposes, such as historical teaching and learning, or for enhancing museum visits. At the same time, game development has been fuelled by dramatic advances in computer graphics hardware – in turn driven by the success of video games – which have led to a rise in the quality of real-time computer graphics and increased realism in computer games. The successes of games that cross over into educational gaming – or serious gaming, such as the popular Civilization (although “abstract and ahistorical” [6]) and Total War series of entertainment games, as well as games and virtual worlds that are specifically developed for educational purposes, such as Revolution [61] and the Virtual Egyptian Temple [83], all of which exist within a cultural heritage context, reveal the potential of these technologies to engage and motivate beyond leisure time activities.

The popularity of video games, especially among younger people, makes them an ideal medium for educational purposes [116]. As a result there has been a trend towards the development of more complex, serious games, which are informed by both pedagogical and game-like, fun elements. The term ‘serious games’ describes a relatively new concept, computer games that are not limited to the aim of providing entertainment, that allow for collaborative use of 3D spaces that are used for learning and educational purposes in a number of application domains. Typical examples are game engines and online virtual environments that have been used to design and implement games for non-leisure purposes, e.g. in military and health training [114,204], as well as cultural heritage (Figure 1).



Fig. 1 ‘Roma Nova’ - Experiencing ‘Rome Reborn’ as a game.

This report explores the wider research area of interactive games and related applications with a cultural heritage context and the technologies used for their creation. Modern games technologies (and related optimisations [27]) allow the real-time interactive visualisation/simulation of realistic virtual heritage scenarios, such as reconstructions of ancient sites and monuments, while using relatively basic consumer machines. Our aim is to provide an overview of the methods and techniques used in entertainment games that can potentially be deployed in cultural heritage contexts,

as demonstrated by particular games and applications, thus making cultural heritage much more accessible.

Serious games can exist in the form of mobile applications, simple web-based solutions, more complex ‘mashup’ applications (e.g. combinations of social software applications) or in the shape of ‘grown-up’ computer games, employing modern games technologies to create virtual worlds for interactive experiences that may include socially based interactions, as well as mixed reality games that combine real and virtual interactions, all of which can be used in cultural heritage applications. This state-of-the-art report focusses on the serious games technologies that can be found in modern computer games.

The report is divided into two main sections:

- The first of these is concerned with the area of cultural heritage and serious games, which integrate the core technologies of computer games with principled pedagogical methodologies. This is explored in a range of characteristic case studies, which include entertainment games that can be used for non-leisure purposes as well as virtual museums and educationally focused and designed cultural heritage projects.
- The second part investigates those computer games technologies that are potentially useful for the creation of cultural heritage games, such as real-time rendering techniques, mixed reality technologies and subdomains of (game) artificial intelligence. This literature review includes discussions of strengths and weaknesses of the most prominent methods, indicating potential uses for cultural heritage serious games and illustrating challenges in their application.

2 The State-of-the-Art in Serious Games

The state-of-the-art in Serious Game technology is identical to the state-of-the-art in Entertainment Games technology. Both types of computer game share the same infrastructure, or as Zyda notes, “applying games and simulations technology to non-entertainment domains results in serious games” [204]. The main strengths of serious gaming applications may be generalised as being in the areas of communication, visual expression of information, collaboration mechanisms, interactivity and entertainment.

Over the past decade there have been tremendous advances in entertainment computing technology, and “today’s games are exponentially more powerful and sophisticated than those of just three or four years ago” [168], which in turn is leading to very high consumer expectations. Real-time computer graphics can achieve near-photorealism and virtual game worlds are usually populated with considerable amounts of high quality content, creating a rich user experience. In this respect, Zyda [204] argues that while pedagogy is an implicit component of a serious game, it should be secondary to entertainment, meaning that a serious game that is not ‘fun’ to play would be useless, independent of its pedagogical content or value. This view is not shared by all, and there exist design methodologies for the development of games incorporating pedagogic elements, such as the four dimensional framework [62], which outlines the centrality of four elements that can be used as design and evaluation criteria for the creation of serious games. In any case there is a need for the game developers and instructional designers to work together to develop engaging and motivating serious games for the future.

2.1 Online Virtual Environments

There is a great range of different online virtual world applications – at least 80 virtual world applications existed in 2008 with another 100 planned for 2009. The field is extensive, not just in terms of potential use for education and training but also in terms of actual usage and uptake by users, which is amply illustrated by the online platform Second Life (Linden Labs), which currently has 13 million registered accounts worldwide. The use of Second Life for supporting seminar activities, lectures and other educational purposes has been documented in a number of recent reports and a wide range of examples of Second Life use by UK universities has been documented [93]. Online virtual worlds provide excellent capabilities for creating effective distance and online learning opportunities through the provision of unique support for distributed groups (online chat, the use of avatars, document sharing etc.). This benefit has so far been most exploited in business where these tools have been used to support distributed or location-independent working groups or communities [86]. Online virtual worlds in this way facilitate the development of new collaborative models for bringing together subject matter experts and tutors from around the world, and in terms of learning communities are opening up opportunities for learning in international cohorts where students from more than one country or location can learn in mixed reality contexts including classroom and non-classroom based groups (<https://lg3d-wonderland.dev.java.net>). Online virtual worlds also notably offer real opportunities for training, rehearsing and role playing.

2.2 Application to Cultural Heritage: Case Studies

This section provides an overview of some of the most characteristic case studies in cultural heritage. In particular the case studies have been categorised into three types of computer-game-like applications including: prototypes and demonstrators; virtual museums; and commercial historical games.

2.2.1 Prototypes and Demonstrators

The use of visualisation and virtual reconstruction of ancient historical sites is not new, and a number of projects have used this approach to study crowd modelling [7, 115]. Several projects are using virtual reconstructions in order to train and educate their users. Many of these systems have, however, never been released to the wider public, and have only been used for academic studies. In the following section the most significant and promising of these are presented.

Roma Nova

The Rome Reborn project is the world's largest digitisation project and has been running for 15 years. The main aims of the project are to produce a high resolution version of Rome at 320 AD (Figure 2), a lower resolution model for creating a 'mashup' application with 'Google Earth' (<http://earth.google.com/rome/>), and finally the collaborative mode of the model for use with virtual world applications and aimed primarily at education [63].

In order to investigate the efficacy of the Rome Reborn Project for learning, exploration, re-enactment and research of cultural and architectural aspects of ancient

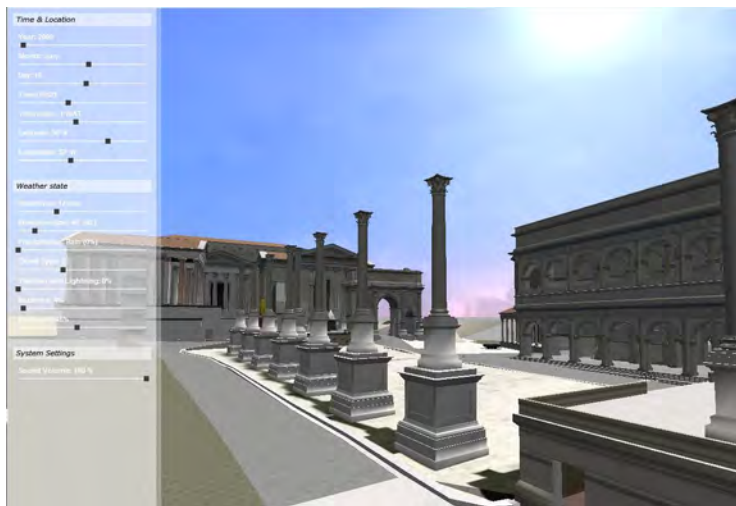


Fig. 2 ‘Rome Reborn’ Serious Game.

Rome the serious game ‘Roma Nova’ is currently under development. In particular, the project aims at investigating the suitability of using this technology to support the archaeological exploration of historically accurate societal aspects of Rome’s life, with an emphasis on political, religious and artistic expressions.

To achieve these objectives, the project will integrate four cutting-edge virtual world technologies with the Rome Reborn model, the most detailed three-dimensional model of Ancient Rome available. These technologies include:

- the Quest3D visualisation engine [70]
- Instinct(maker) artificial life engine (Toulouse University) [165]
- ATOM Spoken Dialogue System
(<http://www.agilingua.com>)
- High resolution, motion captured characters and objects from the period (Red Bedlam).

The use of the Instinct artificial life engine enables coherent crowd animation and therefore the population of the city of Rome with behaviour driven virtual characters. These virtual characters with different behaviours can teach the player about different aspects of life in Rome (living conditions, politics, military) [165]. Agilingua ATOM’s dialogue management algorithm allows determining how the system will react: asking questions, making suggestions, and/or confirming an answer.

This project aims to develop a researchers’ toolkit for allowing archaeologists to test past and current hypotheses surrounding architecture, crowd behaviour, social interactions, topography and urban planning and development, using Virtual Rome as a test-bed for reconstructions. By using such game the researches will be able to analyse the impact of major events. For example, the use of this technique would allow researchers to analyse the impact of major events, such as grain distribution or the influx of people into the city. The experiences of residents and visitors as they pass through and interact with the ancient city can also be explored.

Ancient Pompeii

Pompeii was a Roman city, which was destroyed and completely buried in the first recorded eruption of the volcano Mount Vesuvius in 79 AD [151,152]. For this project a model of ancient Pompeii was constructed using procedural methods [130] and subsequently populated with avatars in order to simulate life in Pompeii in real-time. The main goal of this project was to simulate a crowd of virtual Romans exhibiting realistic behaviours in a reconstructed district of Pompeii [115]. The virtual entities can navigate freely in several buildings in the city model and interact with their environment [7].

Parthenon Project

The Parthenon Project is a short computer animation that “visually reunites the Parthenon and its sculptural decorations” [37]. The Parthenon itself is an ancient monument, completed in 437 BC, and stands in Athens while many of its sculptural decorations reside in the collection of the British Museum, London (UK). The project goals were to create a virtual version of the Parthenon and its separated sculptural elements so that they could be reunited in a virtual representation.

The project involved capturing digital representations of the Parthenon structure and the separate sculptures, recombining them and then rendering the results. The structure was scanned using a commercial laser range scanner, while the sculptures were scanned using a custom 3D scanning system that the team developed specifically for the project [186]. The project made heavy use of image-based lighting techniques, so that the structure could be relit under different illumination conditions within the virtual representation. A series of photographs were taken of the structure together with illumination measurements of the scene’s lighting. An inverse global illumination technique was then applied to effectively ‘remove’ the lighting. The resulting “lighting-independent model” [38] could then be relit using any lighting scheme desired [187, 38].

Although the Parthenon Project was originally an offline-rendered animation, it has since been converted to work in real-time [166,81]. The original Parthenon geometry represented a large dataset consisting of 90-million polygons (after post-processing), which was reduced to 15-million for the real-time version and displayed using dynamic level-of-detail techniques. Texture data consisted of 300MB and had to be actively managed and compressed, while 2.1GB of compressed High-Dynamic-Range (HDR) sky maps were reduced in a pre-processing step. The reduced HDR maps were used for lighting and the extracted sun position was used to cast a shadow map.

2.2.2 Virtual Museums

Modern interactive virtual museums using games technologies [87,100] provide a means for the presentation of digital representations for cultural heritage sites [47] that entertain and educate visitors [73] in a much more engaging manner than was possible only a decade ago. A recent survey paper that examines all the technologies and tools used in museums was recently published [182]. Here we present several examples of this type of cultural heritage serious game, including some virtual museums that can be visited in real-world museums.

Virtual Egyptian Temple

This game depicts a hypothetical Virtual Egyptian Temple [83, 189], which has no real-world equivalent. The temple embodies all of the key features of a typical New Kingdom period Egyptian temple in a manner that an untrained audience can understand. This Ptolemaic temple is divided into four major areas, each one of which houses an instance of the High Priest, a pedagogical agent. Each area of this virtual environment represents a different feature from the architecture of that era.

The objective of the game ‘Gates of Horus’ [82] is to explore the model and gather enough information to answer the questions asked by the priest (pedagogical agent). The game engine that this system is based on is the Unreal Engine 2 (Figure 3) [84], existing both as an Unreal Tournament 2004 game modification [194] for use at home, as well as in the form of a Cave Automatic Virtual Environment (CAVE [35]) system in a real museum.



Fig. 3 New Kingdom Egyptian Temple game.

The Ancient Olympic Games

The Foundation of the Hellenic World has produced a number of gaming applications associated with the Olympic Games in ancient Greece [66]. For example, in the ‘Olympic Pottery Puzzle’ exhibit the user must re-assemble a number of ancient vases putting together pot shards. The users are presented with a colour-coded skeleton of the vessels with the different colours showing the correct position of the pieces. They then try to select one piece at a time from a heap and place it in the correct position on the vase. Another game is the ‘Feidias Workshop’ which is a highly interactive virtual experience taking place at the construction site of the 15-meter-tall golden ivory statue of Zeus, one of the seven wonders of the ancient world. The visitors enter the two-storey-high workshop and come into sight of an accurate reconstruction of an unfinished version of the famous statue of Zeus and walk among the sculptor’s tools, scaffolding, benches, materials, and moulds used to construct it. They take the role of the sculptor’s assistants and actively help finish the creation of the huge statue, by using virtual tools to apply the necessary materials onto the statue, process the

ivory and gold plates, apply them onto the wooden supporting core and add the finishing touches. Interaction is achieved using the navigation wand of the Virtual Reality (VR) system, onto which the various virtual tools are attached. Using these tools the user helps finish the work on the statue, learning about the procedures, materials and techniques applied for the creation of these marvellous statues. The last example is the



Fig. 4 Walk through Ancient Olympia [66].

‘Walk through Ancient Olympia’, where the user, apart from visiting the historical site, learns about the ancient games themselves by interacting with athletes in the ancient game of pentathlon (Figure 4). The visitors can wonder around and visit the buildings and learn their history and their function: the Heraion, the oldest monumental building of the sanctuary dedicated to the goddess Hera, the temple of Zeus, a model of a Doric peripteral temple with magnificent sculpted decoration, the Gymnasium, which was used for the training of javelin throwers, discus throwers and runners, the Palaestra, where the wrestlers, jumpers and boxers trained, the Leonidaion, which was where the official guests stayed, the Bouleuterion, where athletes, relatives and judges took a vow that they would uphold the rules of the Games, the Treasuries of various cities, where valuable offerings were kept, the Philippeion, which was dedicated by Philip II, king of Macedonia, after his victory in the battle of Chaeronea in 338 BC and the Stadium, where most of the events took place. Instead of just observing the games the visitors take place in them. They can pick up the discus or the javelin and they try their abilities in throwing them towards the far end of the stadium. Excited about the interaction they ask when they will be able to interact with the wrestler one on one. A role-playing model of interaction with alternating roles was tried here with pretty good success as the visitors truly immersed in the environment wish they could participate in more games [66].

Virtual Priory Undercroft

Located in the heart of Coventry, UK, the Priory Undercrofts are the remains of Coventry's original Benedictine monastery, dissolved by Henry VIII. Although archaeologists revealed the architectural structure of the cathedral, the current site is not easily accessible for the public. Virtual Priory Undercroft offers a virtual exploration of the site in both online and offline configurations.

Furthermore, a first version of a serious game (Figure 5) has been developed at Coventry University, using the Object-Oriented Graphics Rendering Engine (OGRE) [198]. The motivation is to raise the interest of children in the museum, as well as cultural heritage in general. The aim of the game is to solve a puzzle by collecting medieval objects that used to be located in and around the Priory Undercroft. Each time a new object is found, the user is prompted to answer a question related to the history of the site. A typical user-interaction might take the form of: "What did St. George slay? – Hint: It is a mythical creature. – Answer: The Dragon", meaning that the user then has to find the Dragon.



Fig. 5 Priory Undercroft – a Serious Game

2.2.3 Commercial Historical Games

Commercial games with a cultural heritage theme are usually of the 'documentary game' [23] genre that depict real historical events (frequently wars and battles), which the human player can then partake in. These are games that were primarily created for entertainment, but their historical accuracy allows them to be used in educational settings as well.

History Line: 1914-1918

An early representative of this type of game was History Line: 1914-1918 (Blue Byte, 1992), an early turn-based strategy game depicting the events of the First World War. The game was realised using the technology of the more prominent game Battle Isle, providing players with a 2D top-down view of the game world, divided into hexagons

that could be occupied by military units, with the gameplay very much resembling traditional board-games.

The game's historical context was introduced in a long (animated) introduction, depicting the geo-political situation of the period and the events leading up to the outbreak of war in 1914. In between battles the player is provided with additional information on concurrent events that shaped the course of the conflict, which is illustrated with animations and newspaper clippings from the period.



Fig. 6 Great Battles of Rome.

Great Battles of Rome

More recently a similar approach was used by the History Channel's Great Battles of Rome (Slitherine Strategies, 2007), another 'documentary game', which mixes interactive 3D real-time tactical simulation of actual battles with documentary information (Figure 6), including footage originally produced for TV documentaries, that places the battles in their historical context.

Total War

The most successful representatives of this type of historical game are the games of the Creative Assembly's Total War series, which provide a gameplay combination of turn-based strategy (for global events) and real-time tactics (for battles). Here, a historical setting is enriched with information about important events and developments that occurred during the timeframe experienced by the player. While the free-form campaigns allow the game's players to change the course of history, the games also include several independent battle-scenarios with historical background information that depict real events and allow players to partake in moments of historical significance.

The use of up-to-date games technology for rendering, as well as the use of highly detailed game assets that are reasonably true to the historical context, enables a fairly

realistic depiction of history. As a result, games from the Total War series have been used to great effect in the visualisation of armed conflicts in historical programmes produced for TV [196].



Fig. 7 Reliving the battle of Brandywine Creek [124] in ‘Empire: Total War’.

The latest titles in the series, ‘Empire: Total War’ (released in 2009), depicting events from the start of the 18th century to the middle of the 19th century, and ‘Napoleon: Total War’ (released in 2010), depicting European history during the Napoleonic Wars, make use of some of the latest developments in computer games technology (Figure 7). The games’ renderer is scalable to support different types of hardware, including older systems, especially older graphics cards (supporting the programmable Shader Model 2), but the highest visual fidelity is only achieved on recent systems (Shader Model 3 graphics hardware) [67]. If the hardware allows for this, shadows for added realism in the virtual world are generated using Screen Space Ambient Occlusion [128, 11], making use of existing depth-buffer information in rendered frames. Furthermore the virtual world of the game is provided with realistic vegetation generated by the popular middleware system SpeedTree (Interactive Data Visualization, Inc.), which “features realistic tree models and proves to be able to visualise literally thousands of trees in real-time” [64]. As a result the human player is immersed in the historical setting, allowing the player to re-live history.

3 The Technology of Cultural Heritage Serious Games

Modern interactive virtual environments are usually implemented using game engines, which provide the core technology for the creation and control of the virtual world. A game engine is an open, extendable software system on which a computer game or a similar application can be built. It provides the generic infrastructure for game creation [204], i.e. I/O (input/output) and resource/asset management facilities. The possible components of game engines include, but are not limited to: rendering engine, audio engine, physics engine, animation engine.

3.1 Virtual World System Infrastructure

The shape that the infrastructure for a virtual environment takes is dictated by a number of components, defined by function rather than organisation, the exact selection of which determines the tasks that the underlying engine is suitable for. A game engine does not provide data or functions that could be associated with any game or other application of the game engine [202]. Furthermore, a game engine is not just an API (Application Programming Interface), i.e. a set of reusable components that can be transferred between different games, but also provides a glue layer that connects its component parts. It is this glue layer that sets a game engine apart from an API, making it more than the sum of its components and sub-systems.

Modern game engines constitute complex parallel systems that compete for limited computing resources [18]. They “provide superior platforms for rendering multiple views and coordinating real and simulated scenes as well as supporting multiuser interaction” [102], employing advanced graphics techniques to create virtual environments. Anderson et al. [4] provide a discussion of several challenges and open problems regarding game engines, which include the precise definition of the role of content creation tools in the game development process and as part of game engines, as well as the identification of links between game genres and game engine architecture, both of which play a crucial role in the process of selecting an appropriate game engine for a given project.

Frequently, the technology used for the development of virtual environments, be they games for entertainment, serious games or simulations, is limited by the development budget. Modern entertainment computer games frequently require “a multimillion-dollar budget” [143] that can now rival the budgets of feature film productions, a significant proportion of which will be used for asset creation (such as 3D models and animations). Some of these costs can be reduced through the use of procedural modelling techniques for the generation of assets, including terrain [136], vegetation [106] or whole urban environments [192]. Game developers are usually faced with the choice of developing a proprietary infrastructure, i.e. their own game engine, or to use an existing engine for their virtual world application. Commercially developed game engines are usually expensive, and while there are affordable solutions, such as the Torque game engine which is favoured by independent developers and which has been successfully used in cultural heritage applications [97,118], these generally provide fewer features, thus potentially limiting their usefulness. If one of the project’s requirements is the use of highly realistic graphics with a high degree of visual fidelity, this usually requires a recent high-end game engine, the most successful of which usually come at a very high licensing fee.

There are alternatives, however, as several older commercially developed engines have been released under Open Source licences, such as the Quake 3 engine (id Tech 3) [177,198], making them easily accessible, and while they do not provide the features found in more recently published games, they nevertheless match the feature sets of the cheaper commercial engines. Furthermore, there exist open source game engines such as the Nebula Device [157], or engine components, such as OGRE [157,198] or ODE (Open Dynamics Engine) [112], which are either commercially developed or close to commercial quality, making them a viable platform for the development of virtual worlds, although they may lack the content creation tools that are frequently packaged with larger commercial engines.

Finally, there is the possibility of taking an existing game and modifying it for one's own purposes, which many recent games allow users to do [194,177]. This has the benefit of small up-front costs, as the only requirement is the purchase of a copy of the relevant game, combined with access to high-spec modern game engines, as well as the content development tools that they contain. Examples for this are the use of the game Civilization III for the cultural heritage game The History Game Canada (<http://historycanadagame.com>) or the use of the Unreal Engine 2 [177] for the development of an affordable CAVE [84], which has been used successfully in cultural heritage applications [83].

3.2 Virtual World User Interfaces

There are different types of interface that allow users to interact with virtual worlds. These fall into several different categories, such as VR and Augmented Reality (AR), several of which are especially useful for cultural heritage applications, and which are presented in this section.

3.2.1 Mixed Reality Technologies

In 1994, Milgram [126] tried to depict the relationship between VR and AR. To illustrate this he introduced two new terms called Mixed Reality (MR), which is a type of VR but has a wider concept than AR, [184] and Augmented Virtuality (AV). On the left hand side of the Reality-Virtuality continuum, there is the representation of the real world and on the right hand side there is the ultimate synthetic environment. MR stretches out in-between these environments and it can be divided into two sub-categories: AR and AV [126]. AR expands towards the real world and thus it is less synthetic than AV which expands towards virtual environments. To address the problem from another perspective a further distinction has been made. This refers to all the objects that form an AR environment: real objects and virtual objects. Real objects are these, which always exist no matter what the external conditions may be. On the other hand, a virtual object depends on external factors but mimics objects of reality. Some of the most interesting characteristics that distinguish virtual objects, which include holograms and mirror images, and real objects are illustrated below [126].

The most obvious difference is that a virtual object can only be viewed through a display device after it has been generated and simulated. Real-world objects that exist in essence, on the contrary, can be viewed directly and / or through a synthetic device. Another factor is the quality of viewed images that are generated using state-of-the-art technologies. Virtual information cannot be sampled directly but must be synthesised, therefore, depending on the chosen resolution, displayed objects may appear real, but their appearance does not guarantee that the objects are real. Virtual and real information may be distinguished depending on the luminosity of the location that it appears in. Images of real-world objects receive lighting information from the position at which they appear to be located while virtual objects do not necessarily, unless the virtual scene is lit exactly like the real-world location in which objects appear to be displayed. This is true for directly viewed real-world objects, as well as displayed images of indirectly viewed objects.

3.2.2 Virtual Reality

Ivan Sutherland originally introduced the first Virtual Reality (VR) system in the 1960s [181]. Nowadays VR is moving from the research laboratories to the working environment by replacing ergonomically limited HMD's (Head-Mounted Displays) with projective displays (such as the well known CAVE and Responsive Workbench) as well as online VR communities. In a typical VR system the user's natural sensory information is completely replaced with digital information. The user's experience of a computer-simulated environment is called immersion. As a result, VR systems can completely immerse a user inside a synthetic environment by blocking all the signals of the real world. In addition, a VR simulated world does not always have to obey all laws of nature. In immersive VR systems, the most common problems of VR systems are of emotional and psychological nature including motion sickness, nausea, and other symptoms, which are created by the high degree of immersiveness of the users.

Moreover, internet technologies have the tremendous potential of offering virtual visitors ubiquitous access via the World Wide Web (WWW) to online virtual environments. Additionally, the increased efficiency of Internet connections (i.e. ADSL / broadband) makes it possible to transmit significant media files relating to the artefacts of virtual museum exhibitions. The most popular technology for WWW visualisation includes Web3D which offers tools such as the Virtual Reality Modeling Language (VRML – <http://www.web3d.org/x3d/vrml/>) and its successor X3D (<http://www.web3d.org/x3d/>), which can be used for the creation of an interactive virtual museum. Many cultural heritage applications based on VRML have been developed for the web [68,145,176]. Another 3D graphics format, is COLLABorative Design Activity (COLLADA – <https://collada.org>) which defines an open standard XML schema (<http://www.w3.org/XML/Schema>) for exchanging digital assets among various graphics software applications that might otherwise store their assets in incompatible formats. One of the main advantages of COLLADA is that it includes more advanced physics functionality such as collision detection and friction (which Web3D does not support).

In addition to these, there are more powerful technologies that have been used in museum environments, which include the OpenSceneGraph (OSG) high performance 3D graphics toolkit (<http://www.openscenegraph.org/projects/osg>) and a variety of 3D game engines. OSG is a freely available (open source) multi-platform toolkit, used by museums [24,110] to generate more powerful VR applications, especially in terms of immersion and interactivity since it supports the integration of text, video, audio and 3D scenes into a single 3D environment. An alternative to OpenSceneGraph, is OpenSG which is an open-source scene graph system used to create real-time VR applications (<http://www.opensg.org/>) On the other hand, 3D game engines are also very powerful and they provide superior visualisation and physics support. Both technologies (OSG and 3D game engines), compared to VRML and X3D, can provide very realistic and immersive museum environments but they have two main drawbacks. First, they require advanced programming skills in order to design and implement custom applications. Secondly, they do not have support for mobile devices such as PDAs and Third Generation mobile phones.

3.2.3 Augmented Reality

The concept of AR is the opposite of the closed world of virtual spaces [183] since users can perceive both virtual and real information. Compared to VR systems, most AR systems use more complex software approaches, usually including some form of computer vision techniques [60] for sensing the real world. The basic theoretical principle is to superimpose digital information directly into a user's sensory perception [55], rather than replacing it with a completely synthetic environment as VR systems do. An interesting point is that both technologies – AR and VR – may process and display the same digital information and that they often make use of identical dedicated hardware. Although AR systems are influenced by the same factors the amount of influence is much less than in VR since only a portion of the environment is virtual. However, there is still a lot of research to be done in AR [8,9,107] to measure accurately its effects on humans.

The requirements related to the development of AR applications in the cultural heritage field have been well documented [20,104,182]. An interactive concept is the Meta-Museum visualised guide system based on AR, which tries to establish scenarios and provide a communication environment between the real world and cyberspace [117]. Another AR system that could be used as an automated tour guide in museums is the automated tour guide, which superimposes audio in the world based on the location of the user [12]. There are many ways in which archaeological sources can be used to provide a mobile AR system. Some of the wide range of related applications includes the initial collection of data to the eventual dissemination of information [163]. MARVINS is an AR assembly, initially designed for mobile applications and can provide orientation and navigation possibilities in areas, such as science museums, art museums and other historic or cultural sites. Augmented information like video, audio and text is relayed from a server via the transmitter-receiver to a head-mounted display [167].

In addition, a number of EU projects have been undertaken in the field of virtual heritage. The SHAPE project [73] combined AR and archaeology to enhance the interaction of persons in public places like galleries and museums by educating visitors about artefacts and their history. The 3DMURALE project [33] developed 3D multimedia tools to record, reconstruct, encode and visualise archaeological ruins in virtual reality using as a test case the ancient city of Sagalassos in Turkey. The Ename 974 project [150] developed a non-intrusive interpretation system to convert archaeological sites into open-air museums, called TimeScope-1 based on 3D computer technology originally developed by IBM, called TimeFrame. ARCHEOGUIDE [180] provides an interactive AR guide for the visualisation of archaeological sites based on mobile computing, networking and 3D visualisation providing the users with a multi-modal interaction user interface. A similar project is LIFEPLUS [144], which explores the potential of AR so that users can experience a high degree of realistic interactive immersion by allowing the rendering of realistic 3D simulations of virtual flora and fauna (humans, animals and plants) in real-time.

AR technologies can be combined with existing game engine subsystems to create AR game engines [111] for the development of AR games. AR has been applied successfully to gaming in cultural heritage. One of the earliest examples is the Virtual Showcase [14] which is an AR display device that has the same form factor as a real showcase traditionally used for museum exhibits and can be used for gaming. The potentials of AR interfaces in museum environments and other cultural heritage institutions [103] as well as outdoor heritage sites [193] have been also briefly explored for

potential educational applications. A more specific gaming example are the MAGIC and TROC systems [158] which were based on a study of the tasks of archaeological fieldwork, interviews and observations in Alexandria. This takes the form of a mobile game in which the players discover archaeological objects while moving.

Another cultural heritage AR application is the serious game SUA that was part of the BIDAIZERA project [105]. This project takes the form of a play which recreates the 1813 battle between the English and the French in San Sebastian. Researchers developed an interactive system based on AR and VR technologies for recreational and educational applications with tourist, cultural and socio-economical contents, the prototype for which was presented at the Museo del Monte Urgull in San Sebastian.

3.3 Advanced Rendering Techniques

One of the most important elements of the creation of interactive virtual environments is the visual representation of these environments. Although serious games have design goals that are different from those of pure entertainment video games, they can still make use of the wide variety of graphical features and effects that have been developed in recent years. The state-of-the-art in this subject area is broad and, at times, it can be difficult to specify exactly where the ‘cutting edge’ of the development of an effect lies. A number of the techniques that are currently in use were originally developed for offline applications and have only recently become adopted for use in real-time applications through improvements in efficiency or hardware. Here, the ‘state-of-the-art’ for real-time lags several years behind that for offline – good examples of this would be raytracing or global illumination, which we shall briefly examine. A number of effects, however, are developed specifically for immediate deployment on current hardware and can make use of specific hardware features – these are often written by hardware providers themselves to demonstrate their use or, of course, by game developers. Other real-time graphical features and effects can be considered to follow a development cycle, where initially they are proven in concept demonstrations or prototypes, but are too computationally expensive to implement in a full application or game. Over time these techniques may then progressively be optimised for speed, or held back until the development of faster hardware allows their use in computer games.

The primary reason for the proliferation of real-time graphics effects has been due to advances in low-cost graphics hardware that can be used in standard PCs or games consoles. Modern graphics processing units (GPUs) are extremely powerful parallel processors and the graphics pipeline is becoming increasingly flexible. Through the use of programmable shaders, which are small programs that define and direct part of the rendering process, a wide variety of graphical effects are now possible for inclusion in games and virtual environments, while there also exist a range of effects that are currently possible but still too expensive for practical use beyond anything but the display of simple scenes.

The graphics pipeline used by modern graphics hardware renders geometry using rasterisation, where an object is drawn as triangles which undergo viewing transformations before they are converted directly into pixels. In contrast, ray-tracing generates a pixel by firing a corresponding ray into the scene and sampling whatever it may hit. While the former is generally faster, especially using the hardware acceleration on modern graphics cards, it is easier to achieve effects such as reflections using ray-tracing. Although the flexibility of modern GPUs can allow ray-tracing [153] in real-time [78,

175], as well as fast ray-tracing now becoming possible on processors used in games consoles [13], rasterisation is currently still the standard technique for computer games.

Although the modern graphics pipeline is designed and optimised to rasterise polygonal geometry, it should be noted that other types of geometry exist. Surfaces may be defined using a mathematical representation, while volumes may be defined using ‘3D textures’ of voxels or, again, using a mathematical formula [48]. The visualisation of volumetric ‘objects’, which are usually semi-opaque, is a common problem that includes features such as smoke, fog and clouds. A wide variety of options exist for rendering volumes [48,26], although these are generally very computationally expensive and it is common to emulate a volumetric effect using simpler methods. This often involves drawing one or more rectangular polygons to which a four-channel texture has been applied (where the fourth, alpha, channel represents transparency) – for example a cloud element or wisp of smoke. These may be aligned to always face the viewer as billboards [1], a common game technique with a variety of uses [197], or a series of these may be used to slice through a full volume at regular intervals. An alternative method for rendering full volumes is ray-marching, where a volume is sampled at regular intervals along a viewing ray, which can now be implemented in a shader [34], or on processors that are now being used in games consoles [92].

It is sometimes required to render virtual worlds, or objects within worlds, that are so complex or detailed that they cannot fit into the graphics memory, or even the main memory, of the computer – this can be especially true when dealing with volume data. Assuming that the hardware cannot be further upgraded, a number of options exist for such rendering problems. If the scene consists of many complex objects at varying distances, it may be possible to adopt a level-of-detail approach [49] and use less complex geometry, or even impostors [1], to approximate distant objects [166]. Alternatively, if only a small sub-section of the world or object is in sight at any one time, it may be possible to hold only these visible parts in memory and ‘stream’ replace them as new parts come into view, which is usually achieved by applying some form of spatial partitioning [34]. This streaming approach can also be applied to textures that are too large to fit into graphics memory [129]. If too much is visible at one time for this to be possible, a cluster of computers may be used, where the entire scene is often too large for a single computer to hold in memory but is able to be distributed among the cluster with the computers’ individual renders being accumulated and composited together [80] or each computer controlling part of a multi-screen tile display [200].

3.3.1 Post-Processing Effects

One important category of graphical effect stems from the ability to render to an off-screen buffer, or even to multiple buffers simultaneously, which can then be used to form a feedback loop. A polygon may then be drawn (either to additional buffers or to the visible framebuffer) with the previously rendered texture(s) made available to the shader. This shader can then perform a variety of ‘post-processing’ effects.

Modern engines frequently include a selection of such effects [56], which can include more traditional image processing, such as colour transformations [21,15], glow [85], or edge-enhancement [135], as well as techniques that require additional scene information such as depth of field [69,203], motion blur [161] and others which will be mentioned in specific sections later.

The extreme of this type of technique is deferred shading, where the entire lighting calculations are performed as a ‘post-process’. Here, the scene geometry is rendered

into a set of intermediate buffers, collectively called the G-buffer, and the final shading process is performed in image-space using the data from those buffers [94].

3.3.2 Transparency, Reflection and Refraction

The modern real-time graphics pipeline does not deal with the visual representation of transparency, reflection or refraction and their emulation must be dealt with using special cases or tricks. Traditionally, transparency has been emulated using alpha blending [1], a compositing technique where a ‘transparent pixel’ is combined with the framebuffer according to its fourth colour component (alpha). The primary difficulty with this technique is that the results are order dependent, which requires the scene geometry to be sorted by depth before it is drawn and transparency can also present issues when using deferred shading [58]. A number of order-independent transparency techniques have been developed, however, such as depth-peeling [53, 132].



Fig. 8 Achieving a mirror effect by rendering the geometry twice [5].

Mirrored background reflections may be achieved using an environment map [17, 197], which can be a simple but effective method of reflecting a static scene. If the scene is more dynamic, but relatively fast to render, reflections on a flat surface may be achieved by drawing the reflective surface as transparent and mirroring the entire scene geometry about the reflection surface, drawing the mirrored geometry behind it (Figure 8) or, for more complex scenes, using reduced geometry methods such as impostors [185]. Alternatively, six cameras can be used to produce a dynamic environment map [19]. Alternative methods have also been developed to address the lack of parallax, i.e. apparent motion offsets due to objects at different distances, which are missing in a fixed environment map [201].

Perhaps surprisingly on first note, simple refraction effects can be achieved using very similar techniques to those used for reflection. The only differences are that the

sample ray direction points inside the object and that it is bent due to the difference in refractive indices of the two materials, in accordance with Snell's Law [1]. Thus, environment mapping can be used for simple refractions in a static scene, which may be expanded to include chromatic dispersion [57]. In some cases, refraction may also be achieved as a post-processing effect [199].

3.3.3 Surface Detail

The simplest method of adding apparent detail to a surface, without requiring additional geometry, is texture mapping. The advent of pixel shaders means that textures can now be used in more diverse ways to emulate surface detail [162,197,1].

A variety of techniques exist for adding apparent high-resolution bump detail to a low-resolution mesh. In normal mapping [16] the texture map stores surface normals, which can then be used for lighting calculations. Parallax mapping [89] uses a surface height map and the camera direction to determine an offset for texture lookups. Relief texture mapping [137,197] is a related technique which performs a more robust ray-tracing of the height map and can provide better quality results at the cost of performance.

3.3.4 Lighting

The old fixed-function graphics pipeline supported a per-vertex Gouraud lighting model [138], but programmable shaders now allow the developer to implement their own lighting model [162,77]. In general, though, the fixed-function lighting equation is split into: a diffuse component, where direct lighting is assumed to be scattered by micro-facets on the surface; a specular component, which appears as a highlight and is dependent on the angle between the viewer and the light; and an ambient component, which is an indirect 'background' lighting component due to light that has bounced off other objects in the scene [1].

Shadows

Although the graphics pipeline did not originally support shadows, it does now provide hardware acceleration for texture samples of a basic shadow map [1,49]. However, this basic method suffers from aliasing issues, is typically low resolution and can only result in hard shadow edges. Except in certain conditions, the majority of shadows in the real world exhibit a soft penumbra, so there is a desire within computer graphics to achieve efficient soft shadows, for which a large number of solutions have been developed [74,10]. Shadowing complex objects such as volumes can also present issues, many of which have also been addressed [109,71,160].

High-Dynamic-Range Lighting

HDR Lighting is a technique that has become very popular in modern games [174,49]. It stems from the fact that real world luminance has a very high dynamic range, which means that bright surface patches are several orders of magnitude brighter than dark surface patches – for example, the sun at noon “may be 100 million times brighter than starlight” [155]. In general, this means that the 8-bit integers traditionally used in each component of the RGB triplet of pixels in the framebuffer, are woefully inadequate for representing real luminance ranges. Thankfully, modern hardware now allows a greater

precision in data types, so that calculations may be performed in 16 or even 32-bit floating-point format, although it should be noted that a performance penalty usually occurs when using more precise formats.

One of the most striking visual effects associated with HDR lighting is bloom, where extremely bright patches appear to glow. Practically, this is usually applied as a post-process effect in a similar way to a glow effect, where bright patches are drawn into a separate buffer which is blurred and then combined with the original image [91, 90]. This can also be applied to low-dynamic-range images, to make them appear HDR [178].

Modern displays still use the traditional 8-bit per colour component format (with a few exceptions [171]), so the HDR floating point results must be converted, which is the process of tonemapping [155]. Some tonemapping methods allow the specification of a brightness, or exposure value as taken from a physical camera analogy. In an environment where the brightness is likely to change dramatically this exposure should be automatically adjusted – much like a real camera does today. Various methods are available to achieve this, such as by downsampling the entire image to obtain the average brightness [91], or by asynchronous queries to build a basic histogram of the brightness level to determine the required exposure [125,169].

Indirect Lighting: Global Illumination

Incident light on a surface can originate either directly from a light source, or indirectly from light reflected by another surface. Global illumination techniques account for both of these sources of light, although in such methods it is the indirect lighting component that is usually of most interest and the most difficult to achieve. The main difficulty is that in order to render a surface patch, the light that is reflected by all other surface patches in the scene must be known. This interdependence can be costly to compute, especially for dynamic scenes, and although indirect lighting accounts for a high proportion of real world illumination, the computational cost of simulating its effects has resulted in very limited use within real-time applications. [45]

The simplest inclusion of indirect lighting is through pre-computed and baked texture maps, which can store anything from direct shadows or ambient occlusion results to those from radiosity or photon mapping [128]. However, this technique is only viable for completely static objects within a static scene. Another simple global illumination technique, which is commonly associated with HDR lighting, is image-based lighting [155]. Here, an environment map stores both direct and indirect illumination as a simple HDR image, which is then used to light objects in the scene. The image may be captured from a real-world location, drawn by an artist as an art asset or generated in a pre-processing stage by sampling the virtual environment. Multiple samples can then be used to light a dynamic character as it moves through the (static) environment [127]. Although the results can be very effective, image-based lighting cannot deal with fully dynamic scenes without having to recompute the environment maps, which may be costly.

Fully dynamic global illumination techniques generally work on reduced or abstracted geometry, such as using discs to approximate the geometry around each vertex for ambient occlusion [172,76]. It is also possible to perform some operations as a post-process, such as ambient occlusion [128] and even approximations for single-bounce indirect lighting [159]. The general-purpose use of the GPU has also allowed for radiosity at near real-time for very small scenes [30] and fast, but not yet real-time, photon mapping [154]. The latter technique can also be used to simulate caustics, which

are bright patches due to convergent rays from a refractive object, in real-time on the GPU [95], although other techniques for specifically rendering caustics are also possible [195], including as an image-space post-process effect [199], or by applying the 'Caustic Cones' that utilise an intensity map generated from real photographic images [88].

3.4 Artificial Intelligence

Another important aspect of the creation of populated virtual environments as used in cultural heritage applications is the creation of intelligent behaviour for the inhabitants of the virtual world, which is achieved using artificial intelligence (AI) techniques.

It is important to understand that when we refer to the AI of virtual entities in virtual environments, that which we refer to is not truly AI – at least not in the conventional sense [120] of the term. The techniques applied to virtual worlds, such as computer games, are usually a mixture of AI related methods whose main concern is the creation of a believable illusion of intelligence [170], i.e. the behaviour of virtual entities only needs to be believable to convey the presence of intelligence and to immerse the human participant in the virtual world.

The main requirement for creating the illusion of intelligence is perception management, i.e. the organisation and evaluation of incoming data from the AI entity's environment. This perception management mostly takes the form of acting upon sensor information but also includes communication between or coordination of AI entities in environments which are inhabited by multiple entities which may have to act co-operatively. The tasks which need to be solved in most modern virtual world applications such as computer games and to which the intelligent actions of the AI entities are usually restricted to (by convention rather than technology) are [2]:

- decision making
- path finding (planning)
- steering (motion control)

The exact range of problems that AI entities within a computer game have to solve depends on the context in which they exist and the virtual environment in which the game takes place. Combs and Ardoint [29] state that a popular method for the implementation of game AI is the use of an 'environment-based programming style', i.e. the creation of the virtual game world followed by the association of AI code with the game world and the entities that exist in it. This means that the AI entity intelligence is built around and is intrinsically linked to the virtual game environment. This type of entity intelligence can be created using 'traditional' methods for 'decision making', 'path finding' and 'steering'.

Of the three common AI tasks named above, 'decision making' most strongly implies the use of intelligence. Finite state machines (FSMs) are the most commonly used technique for implementing decision making in games [65]. They arrange the behaviour of an AI entity in logical states – defining one state per possible behaviour – of which only one, the entity's behaviour at that point in time, is active at any one time. In game FSMs each state is usually associated with a specific behaviour and an entity's actions are often implemented by linking behaviours with pre-defined animation cycles for the AI entity that allow it to enact the selected behaviour [142]. It is relatively simple to program a very stable FSM that may not be very sophisticated but that "will get the job done". The main drawback of FSMs is that they can become very complex and

hard to maintain, while on the other hand the behaviour resulting from a too simple FSM can easily become predictable. To overcome this problem sometimes hierarchical FSMs are used that break up complex states into a set of smaller ones that can be combined, allowing the creation of larger and more complex FSMs.

In recent years, there has been a move towards performing decision making using goal-directed techniques to enable the creation of nondeterministic behaviour. Dybsand describes this as a technique in which an AI entity “will execute a series of actions ... that attempt to accomplish a specific objective or goal” [46]. In its simplest form, goal-orientation can be implemented by determining a goal with an embedded action sequence for a given AI entity. This action sequence, the entity’s plan, will then be executed by the entity to satisfy the goal [140]. Solutions that allow for more diverse behaviour can improve this by selecting an appropriate plan from a pre-computed ‘plan library’ [52] instead of using a built-in plan. More complex solutions use plans that are computed dynamically, i.e. ‘on the fly’, as is the case with Goal-Oriented Action Planning (GOAP) [140]. In GOAP the sequence of actions that the system needs to perform to reach its end-state or goal is generated in real-time by using a planning heuristic on a set of known values which need to exist within the AI entity’s domain knowledge. To achieve this in his implementation of GOAP, Orkin [141] separates the actions and goals, implicitly integrating preconditions and effects that define the planner’s search space, placing the decision making process into the domain of the planner. This can be further improved through augmenting the representation of the search space by associating costs with actions that can satisfy goals, effectively turning the AI entity’s knowledge base into a weighted graph. This then allows the use of path planning algorithms that find the shortest path within a graph as the planning algorithm for the entity’s high-level behaviour [142]. This has the additional benefit of greater code re-use as the planning method for high-level decision making, as well as path planning is the same and can therefore be executed by the same code module [141] if the representations of the search space are kept identical. The most popular path planning algorithm used in modern computer games is the A* (A-Star) algorithm [179, 119, 133], a generalisation of Dijkstra’s algorithm [41]. A* is optimal, i.e. proven to find the optimal path in a weighted graph if an optimal solution exists [39], which guarantees that AI entities will find the least costly path if such a solution exists within the search space.

Challenges in game AI that are relevant to serious games include the construction of intelligent interfaces [108], such as tutoring systems or virtual guides, and particularly real-time strategy game AI, part of which is concerned with the modelling of great numbers of virtual entities in large scale virtual environments. Challenges there include spatial and temporal reasoning [22], which can be addressed through the use of potential fields [72].

3.4.1 Crowd Simulation

The AI techniques described in the previous section are important tools with which more complex systems can be constructed. A domain of great potential relevance to cultural heritage that is derived from such techniques is the simulation of crowds of humanoid characters. If one wishes to reconstruct and visualise places and events from the past, a crowd of real-time virtual characters, if appropriately attired and behaving, can add new depths of immersion and realism to ancient building reconstructions. These characters can feature merely as a backdrop [28] to add life to a reconstruction,

or can assume the centre stage in more active roles, for example, as virtual tour guides to direct the spectator [40]. Indeed, the type of crowd or character behaviour to be simulated varies greatly with respect to the type of scenario that needs to be modelled. In this vein, [191] model crowd behaviour of worshippers in a virtual mosque, while [115] and [164] focus on the creation of more general pedestrian crowd behaviours, the former for populating a virtual reconstruction of a city resembling ancient Rome.

More general crowd synthesis and evaluation techniques are also directly applicable to crowd simulation in cultural heritage. A variety of different approaches have been taken, most notably the use of social force models [75], path planning [96], behavioural models incorporating perception and learning [173] sociological effects [131] and hybrid models [146].

The study of real world corpus has also been used as a basis for synthesising crowd behaviour in approaches that do not entail the definition of explicit behaviour models. Lerner et al. [101] manually track pedestrians from an input video containing real world behaviour examples. They use this data to construct a database of pedestrian trajectories for different situations. At runtime, the database is queried for similar situations matching those of the simulated pedestrians: the closest matching example from the database is selected as the resulting trajectory for each pedestrian and the process is repeated.

Lee et al. [98] simulate behaviours based on aerial-view video recordings of crowds in controlled environments. A mixture of manual annotation and semi-automated tracking provides information from video about individuals' trajectories. These are provided as inputs to an agent movement model that can create crowd behaviours of a similar nature to those observed in the original video.

Human perception of the animation of crowds and characters has been increasingly recognised as an important factor in achieving more realistic simulations. Research has been conducted regarding the perception of animation and motion of individuals [156, 123], groups [50, 121] and crowds [148, 51]. For example, [148] examined the perceptual plausibility of pedestrian orientations and found that participants were able to consistently distinguish between those virtual scenes where the character orientations matched the orientations of the humans in the corresponding real scenes and those where the character orientations were artificially generated, according to a number of different rule types. The results of such perceptual studies can be linked to synthesis, in order to create more credible animations [122].

A key factor of differentiation between crowd control methods concerns where knowledge is stored in the system. One approach is to endow knowledge separately to individual characters, an extreme example of which would create autonomous agents that have their own artificial perceptions, reasoning, memories, etc with respect to the environment, as in [96]. Another method is to place knowledge into the environment itself, to create a shared or partially-shared database accessible to characters. According to this smart object methodology [147], graphical objects are tagged with behavioural information and may inform, guide or even control characters. Such an approach is applicable also to crowd simulation in urban environments. For example, navigation aids, placed inside the environment description, may be added by the designer during the construction process. These have been referred to as annotations [44]. The resulting environment description [54, 188, 149] contains additional geometric, semantic and spatial partitioning information for informing pedestrian behaviour, thus transferring a degree of the behavioural intelligence into the environment. In [79], for example, skeletal splines are defined that are aligned with walkways. These splines, called ribbons,

provide explicit information for groups to use, such as the two major directions of travel on the walkway. In addition to environment annotation and mark-up, interfaces for managing the definition of crowd scenarios have also been investigated. Crowdbush [190] provides an intuitive way for designers to add crowds of characters into an environment using tools analogous to those found in standard 2D painting packages. It allows designers to paint crowds and apply attributes and characteristics using a range of different tools in real-time, obtaining immediate feedback about the results.

3.4.2 Annotated Entities and Environments

A fairly recent method for enabling virtual entities to interact with one another as well as their surroundings is the use of annotated worlds. The mechanism for this, which we refer to using the term ‘Annotated Entities’, has been described using various names, such as ‘Smart Terrain’ [25], ‘Smart Objects’ [147,142] and ‘Annotated Environment’ [43], all of which are generally interchangeable and mostly used with very similar meanings, although slight differences in their exact interpretation sometimes remain. A common aspect to all of the implementations that utilise this mechanism is the indirect approach to the creation of believable intelligent entities.

The idea of annotated environments is a computer application of the theory of affordance (or affordance theory) [32] that was originally developed in the fields of psychology and visual perception. Affordance theory states that the makeup and shape of objects contains suggestions about their usage. Affordance itself is an abstract concept, the implementation of which is greatly simplified by annotations that work like labels containing instructions which provide an explicit interpretation of affordances. Transferred into the context of a virtual world, this means that objects in the environment contain all of the information that an AI controlled entity will need to be able to use them, effectively making the environment ‘smart’.

A beneficial side effect of this use of ‘annotated’ objects [42] is that the complexity of the entities is neutral to the extent of the domain knowledge that is available for their use, i.e. the virtual entities themselves can not only be kept relatively simple, but they do not need to be changed at all to be able to make use of additional knowledge. This allows for the rapid development of game scenarios [32] and if all annotated objects use the same interface to provide knowledge to the world’s entities then there is no limit to the scalability of the system, i.e. the abilities of AI controlled entities can practically be extended indefinitely [139] despite a very low impact on the system’s overall performance. Furthermore, this method provides an efficient solution to the ‘anchoring problem’ [31] of matching sensor data to the symbolic representation of the virtual entity’s knowledge as objects in the world themselves have the knowledge as to how other virtual entities can interact with them.

Annotations have been employed in several different types of applications in order to achieve different effects. They have proven popular for the animation of virtual actors in computer animation production, where they facilitate animation selection [99], i.e. the choice of appropriate animation sequences that fit the environment. Other uses of annotations include the storage of tactical information in the environment for war games and military simulations [36], which is implemented as sensory annotations to direct the virtual entities’ perception of their environment. Probably the most common form of annotations found in real-time simulated virtual environments affects behaviour selection, usually in combination with animation selection [142], i.e. the virtual entity’s

behaviour and its visual representation (animation) are directed by the annotated objects that it uses.

Virtual entities that inhabit these annotated worlds can be built utilising rule-based system based on simple FSMs in combination with a knowledge interface based on a trigger system that allows the entities to ‘use’ knowledge (instructions) for handling the annotated objects. The interaction protocol employed to facilitate the communication between entity and ‘smart’ object needs to enable the object to ‘advertise’ its features to the entities and then allow them to request from the object relevant instructions (annotations) on its usage [113]. The success of this technique is demonstrated by the best-selling computer game *The Sims*, where ‘Smart Objects’ were used for behaviour selection to great effect. Forbus and Wright [59] state that in *The Sims* all game entities, objects as well as virtual characters, are implemented as scripts that are executed in their own threads within a multitasking virtual machine. A similar approach, based on a scripting language that can represent the behaviours of virtual entities, as well as the objects that they can interact with, has been presented more recently by Anderson [3]. These scripting-language based approaches are most likely to provide solutions for the creation of large scale virtual environments, such as the serious game component of the *Rome Reborn* project. This is the automatic generation of AI content [134], which in combination with techniques such as procedural modelling of urban environments [192], will require the integration of the creation of complex annotations with the procedural generation of virtual worlds, automating the anchoring of virtual entities into their environment.

4 Conclusions

The success of computer games, fuelled among other factors by the great realism that can be attained using modern consumer hardware, and the key techniques of games technology that have resulted from this, have given way to new types of games, including serious games, and related application areas, such as virtual worlds, mixed reality, augmented reality and virtual reality. All of these types of application utilise core games technologies (e.g. 3D environments) as well as novel techniques derived from computer graphics, human computer interaction, computer vision and artificial intelligence, such as crowd modelling. Together these technologies have given rise to new sets of research questions, often following technologically driven approaches to increasing levels of fidelity, usability and interactivity.

Our aim has been to use this state-of-the-art report to demonstrate the potential of games technology for cultural heritage applications and serious games, to outline key problems and to indicate areas of technology where solutions for remaining challenges may be found. To illustrate that first we presented some characteristic case studies illustrating the application of methods and technologies used in cultural heritage. Next, we provided an overview of existing literature of relevance to the domain, discussed the strengths and weaknesses of the described methods and pointed out unsolved problems and challenges. It is our firm belief that we are only at the beginning of the evolution of games technology and that there will be further improvements in the quality and sophistication of computer games, giving rise to serious heritage games of greater complexity and fidelity than is now achievable.

5 Acknowledgements

The authors would like to thank the following: The Herbert Art Gallery & Museum (Coventry, UK), Simon Bakkevig, and Lukasz Bogaj. This report includes imagery generated using the Virtual Egyptian Temple, which is a product of PublicVR (<http://publicvr.org>).

References

1. Akenine-Möller, T., Haines, E., Hoffman, N. (2008). *Real-Time Rendering 3rd Edition*. A. K. Peters
2. Anderson, E.F. (2003). *Playing Smart - Artificial Intelligence in Computer Games*. In: *Proceedings of zfxCON03 Conference on Game Development*
3. Anderson, E.F. (2008). *Scripted smarts in an intelligent virtual environment: behaviour definition using a simple entity annotation language*. In: *Future Play '08: Proceedings of the 2008 Conference on Future Play*, pp 185–188
4. Anderson, E.F., Engel, S., McLoughlin, L., Comminos, P. (2008). *The case for research in game engine architecture*. In: *Future Play '08: Proceedings of the 2008 Conference on Future Play*, pp 228–231
5. Anderson, E.F., McLoughlin, L. (2007). *Critters in the classroom: a 3d computer-game-like tool for teaching programming to computer animation students*. In: *SIGGRAPH '07: ACM SIGGRAPH 2007 educators program*, p. 7
6. Apperley, T.H. (2006). *Virtual unaustralia: Videogames and australia's colonial history*. In: *UNAUSTRALIA 2006: Proceedings of the Cultural Studies Association of Australasia's Annual Conference*
7. Arnold, D., Day, A., Glauert, J., Haegler, S., Jennings, V., Kevelham, B., Laycock, R., Magnenat-Thalmann, N., Mam, J., Maupu, D., Papagiannakis, G., Thalmann, D., Yersin, B., Rodriguez-Echavarría, K. (2008). *Tools for populating cultural heritage environments with interactive virtual humans*. In: *Open Digital Cultural Heritage Systems, EPOCH Final Event Rome*
8. Azuma, R. (1997). *A survey of augmented reality*. *Presence: Teleoperators and Virtual Environments* **6**(4): 355–385
9. Azuma, R., Bailiot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B. (2001). *Recent advances in augmented reality*. *IEEE Computer Graphics and Applications* **21**(6): 34–47
10. Bavoil, L. (2008). *Advanced soft shadow mapping techniques*. Presentation at *The Game Developers Conference 2008*
11. Bavoil, L., Sainz, M. (2008). *Screen space ambient occlusion*. NVIDIA developer information: <http://developers.nvidia.com>
12. Bederson, B.B. (1995). *Audio augmented reality: a prototype automated tour guide*. In: *CHI '95: Conference companion on Human factors in computing systems*, pp 210–211
13. Benthin, C., Wald, I., Scherbaum, M., Friedrich, H. (2006). *Ray tracing on the cell processor*. pp 15–23
14. Bimber, O., Frhlich, B., Schmalstieg, D., Encarnao, L.M. (2001). *The virtual showcase*. *IEEE Computer Graphics and Applications* **21**(6): 48–55
15. Bjorke, K. (2004). *Color Controls*. In: R. Fernando (ed) *GPU Gems*, pp 363–373. Pearson Education
16. Blinn, J.F. (1978). *Simulation of wrinkled surfaces*. *SIGGRAPH Comput. Graph.* **12**(3): 286–292
17. Blinn, J.F., Newell, M.E. (1976). *Texture and reflection in computer generated images*. *Commun. ACM* **19**(10): 542–547
18. Blow, J. (2004). *Game development harder than you think*. *ACM Queue* **1**(10): 28–37
19. Blythe, D. (2006). *The direct3d 10 system*. *ACM Trans. Graph.* **25**(3): 724–734
20. Brogni, B., Avizzano, C., Evangelista, C., Bergamasco, M. (1999). *Technological approach for cultural heritage: augmented reality*. In: *RO-MAN '99: Proceedings of the 8th IEEE International Workshop on Robot and Human Interaction*, pp 206–212
21. Burkersroda, R. (2005). *Colour Grading*. In: W. Engel (ed) *Shader X3: Advanced Rendering with DirectX and OpenGL*, pp 357–362. Charles River Media

22. Buro, M. (2004). Call for ai research in rts games. In: Proceedings of the AAAI-04 Workshop on Challenges in Game AI, pp 139–142
23. Burton, J. (2005). News-game journalism: History, current use and possible futures. *Australian Journal of Emerging Technologies and Society* **3**(2): 87–99
24. Calori, L., Camporesi, C., Forte, M., Guidazzoli, A., Pescarin, S. (2005). Openheritage: Integrated approach to web 3d publication of virtual landscape. In: Proceedings of the ISPRS Working Group V/4 Workshop 3D-ARCH 2005: Virtual Reconstruction and Visualization of Complex Architectures
25. Cass, S. (2002). Mind Games. *IEEE Spectrum* **39**(12): 40–44
26. Cerezo, E., Perez-Cazorla, F., Pueyo, X., Seron, F., Sillion, F. (2005). A survey on participating media rendering techniques. *the Visual Computer*
27. Chalmers, A., Debattista, K. (2009). Level of realism for serious games. In: VS-Games 2009: Proceedings of the IEEE Virtual Worlds for Serious Applications First International Conference, pp 225–232
28. Ciechomski, P.D.H., Ulicny, B., Cetre, R., Thalmann, D. (2004). A case study of a virtual audience in a reconstruction of an ancient roman odeon in aphrodisias. In: The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage, VAST (2004)
29. Combs, N., Ardoint, J. (2004). Declarative versus Imperative Paradigms in Games AI. Available from: <http://www.red3d.com/cwr/games/>
30. Coombe, G., Harris, M. (2005). Global Illumination Using Progressive Refinement Radiosity. In: M. Pharr (ed) *GPU Gems 2*, pp 635–647. Pearson Education
31. Coradeschi, S., Saffiotti, A. (1999). Symbolic Object Descriptions to Sensor Data. Problem Statement. *Linköping Electronic Articles in Computer and Information Science* **4**(9)
32. Cornwell, J., O'Brien, K., Silverman, B., Toth, J. (2003). Affordance Theory for Improving the Rapid Generation, Composability, and Reusability of Synthetic Agents and Objects. In: BRIMS 2003: Proceedings of the Twelfth Conference on Behavior Representations in Modeling and Simulation
33. Cosmas, J., Itegaki, T., Green, D., Grabczewski, E., Weimer, F., Van Gool, L., Zalesny, A., Varrintel, D., Leberl, F., Grabner, M., Schindler, K., Karner, K., Gervautz, M., Hynst, S., Waelkens, M., Pollefeys, M., DeGeest, R., Sablatnig, R., Kampel, M. (2001). 3d murale: a multimedia system for archaeology. In: VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage, pp 297–306
34. Crassin, C., Neyret, F., Lefebvre, S., Eisemann, E. (2009). Gigavoxels: ray-guided streaming for efficient and detailed voxel rendering. In: I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games, pp 15–22
35. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R.V., Hart, J.C. (1992). The cave: audio visual experience automatic virtual environment. *Commun. ACM* **35**(6): 64–72
36. Darken, C.J. (2007). Level Annotation and Test by Autonomous Exploration. In: AIIDE 2007: Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference
37. Debevec, P. (2005). Making "The Parthenon". 6th International Symposium on Virtual Reality, Archaeology, and Cultural Heritage
38. Debevec, P., Tchou, C., Gardner, A., Hawkins, T., Poullis, C., Stumpfel, J., Jones, A., Yun, N., Einarsson, P., Lundgren, T., Fajardo, M., Martinez, P.: Estimating surface reflectance properties of a complex scene under captured natural illumination. Tech. rep., University of Southern California, Institute for Creative Technologies (2004)
39. Dechter, R., Pearl, J. (1985). Generalised Best-First Search Strategies and the Optimality of A*. *Journal of the ACM* **32**(3): 505–536
40. DeLeon, V.J. (1999). Vrnd: Notre-dame cathedral: A globally accessible multi-user real time virtual reconstruction. In: Proceedings of Virtual Systems and Multimedia 1999
41. Dijkstra, E.W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik* **1**: 269–271
42. Doyle, P. (1999). Virtual Intelligence from Artificial Reality: Building Stupid Agents in Smart Environments. In: AAAI '99 Spring Symposium on Artificial Intelligence and Computer Games
43. Doyle, P. (2002). Believability through Context. In: AAMAS '02: Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems, pp 342–349
44. Doyle, P., Hayes-Roth, B. (1998). Agents in annotated worlds. In: AGENTS '98: Proceedings of the second international conference on Autonomous agents, pp 173–180

45. Dutr, P., Bekaert, P., Bala, K. (2003). Advanced Global Illumination. A. K. Peters
46. Dybsand, E. (2004). Goal-Directed Behaviour Using Composite Tasks. In: *AI Game Programming Wisdom 2*, pp 237–245. Charles River Media
47. El-Hakim, S., MacDonald, G., Lapointe, J.F., Gonzo, L., Jemtrud, M. (2006). On the Digital Reconstruction and Interactive Presentation of Heritage Sites through Time. In: *International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, pp 243–250
48. Engel, K., Hadwiger, M., Kniss, J.M., Rezk-Salama, C., Weiskopf, D. (2006). *Real-Time Volume Graphics*. A. K. Peters
49. Engel, W., Hoxley, J., Kornmann, R., Suni, N., Zink, J. (2008). Programming vertex, geometry, and pixel shaders. Online book available at: <http://wiki.gamedev.net/>
50. Ennis, C., McDonnell, R., O'Sullivan, C. (2010). Seeing is believing: body motion dominates in multisensory conversations. *ACM Transactions on Graphics* **29**(4)
51. Ennis, C., Peters, C., O'Sullivan, C. (2010). Perceptual effects of scene context and viewpoint for virtual pedestrian crowds. *ACM Transactions on Applied Perception* (**to appear**)
52. Evans, R. (2001). *AI in Computer Games: The Use of AI Techniques in Black & White*. Seminar Notes, available from: <http://www.dcs.qmul.ac.uk/research/logic/seminars/abstract/EvansR01.html>
53. Everitt, C. (2001). Interactive order-independent transparency. NVIDIA Whitepaper
54. Farenc, N., Boulic, R., Thalmann, D. (1999). An informed environment dedicated to the simulation of virtual humans in urban context. *Computer Graphics Forum* **18**(3)
55. Feiner, S. (2002). Augmented reality: A new way of seeing. *Scientific American* **286**(4): 48–55
56. Feis, A. (2007). Postprocessing Effects in Design. In: W. Engel (ed) *Shader X5: Advanced Rendering Techniques*, pp 463–470. Charles River Media
57. Fernando, R., Kilgard, M.J. (2003). *The Cg Tutorial*. Addison Wesley
58. Filion, D., McNaughton, R. (2008). Effects & techniques. In: *SIGGRAPH '08: ACM SIGGRAPH 2008 classes*, pp 133–164
59. Forbus, K.D., Wright, W. (2001). Some notes on programming objects in The SimsTM. Class Notes, available from: <http://qrg.northwestern.edu/papers/papers.html>
60. Forsyth, D.A., Ponce, J. (2002). *Computer Vision: A Modern Approach*. Prentice Hall
61. Francis, R. (2006). Revolution: Learning about history through situated role play in a virtual environment. In: *Proceedings of the American Educational Research Association Conference*
62. de Freitas, S., Oliver, M. (2006). How can exploratory learning with games and simulations within the curriculum be most effectively evaluated? *Computers and Education* **46**: 249–264
63. Frischer, B. (2008). The rome reborn project. how technology is helping us to study history. *OpEd*, November 10, University of Virginia
64. Fritsch, D., Kada, M. (2004). Visualisation using game engines. *ISPRS commission 5* pp 621–625
65. Fu, D., Houlette, R. (2004). The Ultimate Guide to FSMs in Games. In: *AI Game Programming Wisdom 2*, pp 283–302. Charles River Media
66. Gaitatzes, A., Christopoulos, D., Papaioannou, G. (2004). The Ancient Olympic Games: Being Part of the Experience. In: *VAST 2004: The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pp 19–28
67. Gardner, R. (2009). Empire total war - graphics work shop. Available from (the official) Total War blog: <http://blogs.sega.com/totalwar/2009/03/05/empire-total-war-graphics-work-shop/>
68. Gatermann, H. (2000). From vrml to augmented reality via panorama-integration and eai-java. In: *SIGraDi2000 - Construindo (n)o espacio digital (constructing the digital Space)*, pp 254–256
69. Gillham, D. (2007). Real-time Depth-of-Field Implemented with a Postprocessing-Only Technique. In: W. Engel (ed) *Shader X5: Advanced Rendering Techniques*, pp 163–175. Charles River Media
70. Godbersen, H. (2008). Virtual environments for anyone. *IEEE Multimedia* **15**(3): 90–95
71. Hadwiger, M., Kratz, A., Sigg, C., Bühler, K. (2006). Gpu-accelerated deep shadow maps for direct volume rendering. In: *GH '06: Proceedings of the 21st ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, pp 49–52

72. Hagelbäck, J., Johansson, S.J. (2008). The rise of potential fields in real time strategy bots. In: AIIDE 08: Proceedings of the Fourth Artificial Intelligence and Interactive Digital Entertainment Conference, pp 42–47
73. Hall, T., Ciolfi, L., Bannon, L., Fraser, M., Benford, S., Bowers, J., Greenhalgh, C., Hellström, S.O., Izadi, S., Schnädelbach, H., Flintham, M. (2001). The visitor as virtual archaeologist: explorations in mixed reality technology to enhance educational and social interaction in the museum. In: VAST '01: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage, pp 91–96
74. Hasenfratz, J.M., Lapierre, M., Holzschuch, N., Sillion, F. (2003). A survey of real-time soft shadows algorithms
75. Helbing, D., Molnar, P. (1995). Social force model for pedestrian dynamics. *Physical Review E* **51**(5): 4282–4286
76. Hoberock, J., Jia, Y. (2008). High-Quality Ambient Occlusion. In: H. Nguyen (ed) *GPU Gems 3*, pp 257–274. Pearson Education
77. Hoffman, N. (2006). Physically based reflectance for games
78. Horn, D.R., Sugerma, J., Houston, M., Hanrahan, P. (2007). Interactive k-d tree gpu raytracing. In: I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games, pp 167–174
79. Hostetler, T.R. (2002). Controlling steering behavior for small groups of pedestrians in virtual urban environments. Ph.D. thesis, The University of Iowa
80. Humphreys, G., Houston, M., Ng, R., Frank, R., Ahern, S., Kirchner, P.D., Klosowski, J.T. (2002). Chromium: a stream-processing framework for interactive rendering on clusters. *ACM Trans. Graph.* **21**(3): 693–702
81. Isidoro, J.R., Sander, P.V. (2006). Animated skybox rendering and lighting techniques. In: SIGGRAPH '06: ACM SIGGRAPH 2006 Courses, pp 19–22
82. Jacobson, J., Handron, K., Holden, L. (2009). Narrative and content combine in a learning game for virtual heritage. In: *Computer Applications to Archaeology 2009*
83. Jacobson, J., Holden, L. (2005). The virtual egyptian temple. In: *ED-MEDIA: Proceedings of the World Conference on Educational Media, Hypermedia & Telecommunications*
84. Jacobson, J., Lewis, M. (2005). Game engine virtual reality with caveat. *IEEE Computer* **38**(4): 79–82
85. James, G., O'Rourke, J. (2004). Real-Time Glow. In: R. Fernando (ed) *GPU Gems*, pp 343–362. Pearson Education
86. Jones, C. (2005). Who are you? theorising from the experience of working through an avatar. *E-Learning* **2**(4): 414–425
87. Jones, G., Christal, M. (2002). The future of virtual museums: On-line, immersive, 3d environments. Created Realities Group
88. Jr., J.T.K., Fletcher, R.L., Yu, N., Holod, R., Chalmers, A., Badler, N.I. (2009). Recreating Early Islamic Glass Lamp Lighting. In: *VAST09: The 10th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage*, pp 33–40
89. Kaneko, T., Takahei, T., Inami, M., Kawakami, N., Yanagida, Y., Maeda, T., Tachi, S. (2001). Detailed shape representation with parallax mapping. In: *Proceedings of ICAT 2001*, pp 205–208
90. Kawase, M. (2003). Frame buffer postprocessing effects in double-s.t.e.a.l (wreakless). Presentation at The Game Developers Conference 2003
91. Kawase, M. (2004). Practical implementation of high dynamic range rendering. Presentation at The Game Developers Conference 2004
92. Kim, J., Jaja, J. (2009). Streaming model based volume ray casting implementation for cell broadband engine. *Sci. Program.* **17**(1-2): 173–184
93. Kirriemuir, J. (2008). Measuring the impact of second life for educational purposes. Eduserv Foundation, Available from: <http://www.eduserv.org.uk/foundation/sl/uksnapshot052008>
94. Koonce, R. (2008). Deferred Shading in Tabula Rasa. In: H. Nguyen (ed) *GPU Gems 3*, pp 429–457. Pearson Education
95. Krüger, J., Bürger, K., Westermann, R. (2006). Interactive screen-space accurate photon tracing on GPUs. In: *Rendering Techniques (Eurographics Symposium on Rendering - EGSR)*, pp 319–329
96. Lamarche, F., Donikian, S. (2004). Crowd of virtual humans: a new approach for real time navigation in complex and structured environments. *Computer Graphics Forum* **23**(3): 509–518

97. Leavy, B., Wyeld, T., Hills, J., Barker, C., Gard, S. (2007). The ethics of indigenous storytelling: using the torque game engine to support australian aboriginal cultural heritage. In: proceedings of the DiGRA 2007 Conference, pp 24–28
98. Lee, K.H., Choi, M.G., Hong, Q., Lee, J. (2007). Group behavior from video: a data-driven approach to crowd simulation. In: SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp 109–118
99. Lee, K.H., Choi, M.G., Lee, J. (2006). Motion Patches: Building Blocks for Virtual Environments Annotated with Motion Data. In: SIGGRAPH '06: ACM SIGGRAPH 2006 Papers, pp 898–906
100. Lepouras, G., Vassilakis, C. (2004). Virtual museums for all: employing game technology for edutainment. *Virtual Reality* **8**(2): 96–106
101. Lerner, A., Chrysanthou, Y., Dani, L. (2007). Crowds by example. *Computer Graphics Forum* **26**(3): 655–664
102. Lewis, M., Jacobson, J. (2002). Game engines in scientific research. *Communications of the ACM* **45**(1): 27–31
103. Liarokapis, F. (2007). An augmented reality interface for visualising and interacting with virtual content. *Virtual Reality* **11**(1): 23–43
104. Liarokapis, F., Sylaiou, S., Mountain, D. (2008). Personalizing virtual and augmented reality for cultural heritage indoor and outdoor experiences. In: VAST08: The 9th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage, pp 55–62
105. Linaza, M.T., Cobos, Y., Mentxaka, J., Campos, M.K., Penalba, M. (2007). Interactive Augmented Experiences for Cultural Historical Events. In: VAST07: The 8th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage, pp 23–30
106. Lintermann, B., Deussen, O. (1999). Interactive modeling of plants. *IEEE Computer Graphics and Applications* **19**(1): 56–65
107. Livingston, M.A. (2005). Evaluating human factors in augmented reality systems. *IEEE Computer Graphics and Applications* **25**(6): 6–9
108. Livingstone, D., Charles, D. (2004). Intelligent interfaces for digital games. In: Proceedings of the AAAI-04 Workshop on Challenges in Game AI, pp 6–10
109. Lokovic, T., Veach, E. (2000). Deep shadow maps. In: SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp 385–392
110. Looser, J., Grasset, R., Seichter, H., Billinghamurst, M. (2006). Osgart - a pragmatic approach to mr. In: ISMAR 06: 5th IEEE and ACM International Symposium on Mixed and Augmented Reality
111. Lugin, J., Cavazza, M. (2010). Towards ar game engines. In: SEARIS 2010 - 3rd Workshop on Software Engineering and Architecture of Realtime Interactive Systems
112. Macagon, V., Wünsche, B. (2003). Efficient collision detection for skeletally animated models in interactive environments. In: Proceedings of IVCNZ '03, pp 378–383
113. Macedonia, M. (2000). Using Technology and Innovation to Simulate Daily Life. *IEEE Computer* **33**(4): 110–112
114. Macedonia, M. (2002). Games Soldiers Play. *IEEE Spectrum* **39**(3): 32–37
115. Maim, J., Haegler, S., Yersin, B., Mueller, P., Thalmann, D., Van Gool, L. (2007). Populating ancient pompeii with crowds of virtual romans. In: VAST07: The 8th International Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage, pp 109–116
116. Malone, T.W., Lepper, M.R. (1987). Making learning fun: A taxonomy of intrinsic motivations for learning. In: R.E. Snow, M.J. Farr (eds) *Aptitude, learning and instruction: III. Conative and affective process analyses*, pp 223–253. Erlbaum
117. Mase, K., Kadobayashi, R., Nakatsu, R. (1996). Meta-museum: A supportive augmented-reality environment for knowledge sharing. In: ATR Workshop on Social Agents: Humans and Machines, pp 107–110
118. Mateevitsi, V., Sfakianos, M., Lepouras, G., Vassilakis, C. (2008). A game-engine based virtual museum authoring and presentation system. In: DIMEA '08: Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts, pp 451–457
119. Matthews, J. (2002). Basic A* Pathfinding Made Simple. In: *AI Game Programming Wisdom*, pp 105–113. Charles River Media
120. McCarthy, J. (2007). What is Artificial Intelligence. Available from: <http://www-formal.stanford.edu/~jmc/whatisai/whatisai.html>

-
121. McDonnell, R., Ennis, C., Dobbyn, S., O'Sullivan, C. (2009). Talking bodies: Sensitivity to desynchronization of conversations. *ACM Transactions on Applied Perception* **6**(4)
 122. McDonnell, R., Larkin, M., Hernández, B., Rudomin, I., O'Sullivan, C. (2009). Eye-catching crowds: saliency based selective variation. *ACM Transactions on Graphics* **28**(3)
 123. McDonnell, R., Newell, F., O'Sullivan, C. (2007). Smooth movers: perceptually guided human motion simulation. In: *SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp 259–269
 124. McGuire, T.J. (2006). *The Philadelphia Campaign: Volume One: Brandywine and the Fall of Philadelphia*. Stackpole Books
 125. McTaggart, G., Green, C., Mitchell, J. (2006). High dynamic range rendering in valve's source engine. In: *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, p. 7
 126. Milgram, P., Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems* **E77-D**(12): 1321–1329
 127. Mitchell, J., McTaggart, G., Green, C. (2006). Shading in valve's source engine. In: *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pp 129–142
 128. Mittring, M. (2007). Finding next gen: Cryengine 2. In: *SIGGRAPH '07: ACM SIGGRAPH 2007 courses*, pp 97–121
 129. Mittring, M., GmbH, C. (2008). Advanced virtual texture topics. In: *SIGGRAPH '08: ACM SIGGRAPH 2008 classes*, pp 23–51
 130. Müller, P., Vereenoghe, T., Ulmer, A., Van Gool, L. (2005). Automatic reconstruction of roman housing architecture. In: *Recording, Modeling and Visualization of Cultural Heritage*, pp 287–298
 131. Musse, S.R., Thalmann, D. (1997). A model of human crowd behavior: Group inter-relationship and collision detection analysis. In: *Computer Animation and Simulation '97*, pp 39–52
 132. Nagy, Z., Klein, R. (2003). Depth-peeling for texture-based volume rendering. In: *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*, p. 429
 133. Nareyek, A. (2004). Ai in computer games. *ACM Queue* **1**(10): 58–65
 134. Nareyek, A. (2007). Game ai is dead. long live game ai! *IEEE intelligent Systems* **22**(1): 9–11
 135. Nienhaus, M., Döllner, J. (2003). Edge-enhancement - an algorithm for real-time non-photorealistic rendering. *International Winter School of Computer Graphics, Journal of WSCG* **11**(2): 346–353
 136. Noghani, J., Liarokapis, F., Anderson, E.F. (2010). Randomly generated 3d environments for serious games. In: *VS-GAMES 2010: Proceedings of the 2nd International Conference on Games and Virtual Worlds for Serious Applications*, pp 3–10
 137. Oliveira, M.M., Bishop, G., McAllister, D. (2000). Relief texture mapping. In: *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp 359–368
 138. OpenGL Architecture Review Board, Shreiner, D., Woo, M., Neider, J., Davis, T. (2007). *OpenGL Programming Guide*, 6 edn. Addison-Wesley
 139. Orkin, J. (2002). 12 Tips from the Trenches. In: *AI Game Programming Wisdom*, pp 29–35. Charles River Media
 140. Orkin, J. (2004). Applying Goal-Oriented Action Planning to Games. In: *AI Game Programming Wisdom 2*, pp 217–228. Charles River Media
 141. Orkin, J. (2004). Symbolic Representation of Game World State: Toward Real-Time Planning in Games. In: *Proceedings of the AAAI-04 Workshop on Challenges in Game AI*, pp 26–30
 142. Orkin, J. (2006). Three States and a Plan: The A.I. of F.E.A.R. In: *Proceedings of the 2006 Game Developers Conference*
 143. Overmars, M. (2004). Teaching computer science through game design. *IEEE Computer* **37**(4): 81–83
 144. Papagiannakis, G., Ponder, M., Molet, T., Kshirsagar, S., Cordier, F., Magnenat-Thalmann, M., Thalmann, D. (2002). LIFEPLUS: Revival of life in ancient Pompeii. In: *Proceedings of the 8th International Conference on Virtual Systems and Multimedia (VSMM '02)*
 145. Paquet, E., El-Hakim, S., Beraldin, A., Peters, S. (2001). The virtual museum: Virtualisation of real historical environments and artefacts and three-dimensional shape-based searching. In: *VAA'01: Proceedings of the International Symposium on Virtual and Augmented Architecture*, pp 182–193

146. Pelechano, N., Allbeck, J.M., Badler, N.I. (2007). Controlling individual agents in high-density crowd simulation. In: SCA '07: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp 99–108
147. Peters, C., Dobbyn, S., Mac Namee, B., O'Sullivan, C. (2003). Smart Objects for Attentive Agents. In: Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision
148. Peters, C., Ennis, C., McDonnell, R., O'Sullivan, C. (2008). Crowds in context: Evaluating the perceptual plausibility of pedestrian orientations. In: Eurographics 2008 - Short Papers, pp 33–36
149. Peters, C., O'Sullivan, C. (2009). Metroped: A tool for supporting crowds of pedestrian ai's in urban environments. In: Proceedings of the AISB 2009 Convention: AI and Games Symposium, pp 64–69
150. Pletinckx, D., Callebaut, D., Killebrew, A.E., Silberman, N.A. (2000). Virtual-reality heritage presentation at ename. *IEEE MultiMedia* **7**(2): 45–48
151. Plinius Caecilius Secundus, G. (79). Epistulae vi.16. The Latin Library: <http://www.thelatinlibrary.com/pliny.ep6.html>
152. Plinius Caecilius Secundus, G. (79). Epistulae vi.20. The Latin Library: <http://www.thelatinlibrary.com/pliny.ep6.html>
153. Purcell, T.J., Buck, I., Mark, W.R., Hanrahan, P. (2002). Ray tracing on programmable graphics hardware. *ACM Trans. Graph.* **21**(3): 703–712
154. Purcell, T.J., Donner, C., Cammarano, M., Jensen, H.W., Hanrahan, P. (2003). Photon mapping on programmable graphics hardware. In: HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics Hardware, pp 41–50
155. Reinhard, E., Ward, G., Pattanaik, S., Debevec, P. (2006). High Dynamic Range Imaging: Acquisition, Display and Image-Based Lighting. Morgan Kaufmann
156. Reitsma, P.S.A., Pollard, N.S. (2003). Perceptual metrics for character animation: sensitivity to errors in ballistic motion. *ACM Transactions on Graphics* **22**(3): 537–542
157. Rémond, M., Mallard, T. (2003). Rei: An online video gaming platform. In: Proceedings of the 9th International Erlang/OTP User Conference
158. Renevier, P., Nigay, L., Bouchet, J., Pasqualetti, L. (2004). Generic Interaction Techniques for Mobile Collaborative Mixed Systems. In: CADUI 2004: Proceedings of the Fifth International Conference on Computer-Aided Design of User Interfaces, pp 307–320
159. Ritschel, T., Grosch, T., Seidel, H.P. (2009). Approximating dynamic global illumination in image space. In: I3D '09: Proceedings of the 2009 symposium on Interactive 3D graphics and games, pp 75–82
160. Ropinski, T., Kasten, J., Hinrichs, K.H. (2008). Efficient shadows for gpu-based volume raycasting. In: Proceedings of the 16th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG 2008), pp 17–24
161. Rosado, G. (2008). Motion Blur as a Post-Processing Effect. In: H. Nguyen (ed) GPU Gems 3, pp 575–581. Pearson Education
162. Rost, R.J. (2006). OpenGL Shading Language, 2 edn. Addison-Wesley
163. Ryan, N. (2000). Back to reality: augmented reality from field survey to tourist guide. In: Virtual Archaeology between Scientific Research and Territorial Marketing, proceedings of the VAST EuroConference
164. Ryder, G., Flack, P., Day, A. (2005). A framework for real-time virtual crowds in cultural heritage environments. In: N.R. M. Mudge, S. R (eds) Vast 2005, Short Papers Proceedings, pp 108–113
165. Sanchez, S., Balet, O., Luga, H., Duthen, Y. (2004). Vibes, bringing autonomy to virtual characters. In: Proceedings of the Third IEEE International Symposium and School on Advance Distributed Systems, pp 19–30
166. Sander, P.V., Mitchell, J.L. (2006). Out-of-core rendering of large meshes with progressive buffers. In: ACM SIGGRAPH 2006: Proceedings of the conference on SIGGRAPH 2006 course notes, pp 1–18
167. Sanwal, R., Chakaveh, S., Fostirpoulos, K., Santo, H. (2000). Marvins - mobile augmented reality visual navigational system. *European Research Consortium for Informatics and Mathematics (ERCIM News)* **40**: 39–40
168. Sawyer, B. (2002). Serious games: Improving public policy through game-based learning and simulation. Whitepaper for the Woodrow Wilson International Center for Scholars
169. Scheuermann, T., Hensley, J. (2007). Efficient histogram generation using scattering on gpus. In: I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games, pp 33–37

-
170. Scott, B. (2002). The Illusion of Intelligence. In: *AI Game Programming Wisdom*, pp 16–20. Charles River Media
 171. Seetzen, H., Heidrich, W., Stuerzlinger, W., Ward, G., Whitehead, L., Trentacoste, M., Ghosh, A., Vorozcovs, A. (2004). High dynamic range display systems. pp 760–768
 172. Shanmugam, P., Arikan, O. (2007). Hardware accelerated ambient occlusion techniques on gpus. In: *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pp 73–80
 173. Shao, W., Terzopoulos, D. (2005). Autonomous pedestrians. In: *SCA '05: Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp 19–28
 174. Sherrod, A. (2006). High dynamic range rendering using opengl frame buffer objects. In: *Game Programming Gems 6*, pp 529–536. Charles River Media
 175. Shirley, P. (2006). State of the art in interactive ray tracing. In: *ACM SIGGRAPH 2006 Courses*
 176. Sinclair, P., Martinez, K. (2001). Adaptive hypermedia in augmented reality. In: *Proceedings of the 3rd Workshop on Adaptive Hypertext and Hypermedia Systems, ACM Hypertext 2001 Conference*
 177. Smith, S., Trenholme, D. (2008). Computer game engines for developing first-person virtual environments. *Virtual Reality* **12**(3): 181–187
 178. Sousa, T. (2005). Adaptive Glare. In: W. Engel (ed) *Shader X3: Advanced Rendering with DirectX and OpenGL*, pp 349–355. Charles River Media
 179. Stout, B. (2000). The Basics of A* for Path Planning. In: *Game Programming Gems*, pp 254–263. Charles River Media
 180. Stricker, D., Daehne, P., Seibert, F., Christou, I., Almeida, L., Carlucci, R., Ioannidis, N. (2001). Design and Development Issues for ARCHEOGUIDE: An Augmented Reality based Cultural Heritage On-site Guide. In: *icav3d'01: Proceedings of the International Conference on Augmented, Virtual Environments and Three-Dimensional Imaging*, pp 1–5
 181. Sutherland, I.E. (1965). The Ultimate Display. In: *Proceedings of the IFIP Congress*, vol. 2, pp 506–508
 182. Sylaïou, S., Liarokapis, F., Kotsakis, K., Patias, P. (2009). Virtual museums, a survey on methods and tools. *Journal of Cultural Heritage* **10**(4): 520–528
 183. Tamura, H., Yamamoto, H., Katayama, A. (1999). Steps toward seamless mixed reality. In: Y. Ohta, H. Tamura (eds) *Mixed Reality: Merging Real and Virtual Worlds*, pp 59–79. Ohmsha Ltd/Springer-Verlag
 184. Tamura, H., Yamamoto, H., Katayama, A. (2001). Mixed reality: Future dreams seen at the border between real and virtual worlds. *IEEE Computer Graphics and Applications* **21**(6): 64–70
 185. Tatarchuk, N., Isidoro, J. (2006). Artist-directable real-time rain rendering in city environments. In: *Eurographics Workshop on Natural Phenomena*
 186. Tchou, C. (2002). Image-based models: Geometry and reflectance acquisition systems. Master's thesis, University of California at Berkeley (2002)
 187. Tchou, C., Stumpfel, J., Einarsson, P., Fajardo, M., Debevec, P. (2004). Unlighting the Parthenon. In: *SIGGRAPH '04: ACM SIGGRAPH 2004 Sketches*, p. 80
 188. Thomas, G., Donikian, S. (2000). Virtual humans animation in informed urban environments. In: *Computer Animation 2000*, pp 112–119
 189. Troche, J., Jacobson, J. (2010). An exemplar of ptolemaic egyptian temples. In: *CAA 2010: the 38th Conference on Computer Applications and Quantitative Methods in Archaeology*
 190. Ulicny, B., de Heras Ciechomski, P., Thalmann, D. (2004). Crowdbrush: interactive authoring of real-time crowd scenes. In: *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp 243–252
 191. Ulicny, B., Thalmann, D. (2002). Crowd simulation for virtual heritage. In: *Proc. First International Workshop on 3D Virtual Heritage*, pp 28–32
 192. Vanegas, C.A., Aliaga, D.G., Wonka, P., Mller, P., Waddell, P., Watson, B. (2009). Modeling the appearance and behavior of urban spaces. In: *Eurographics 2009 - State of the Art Reports*, pp 1–16
 193. Vlahakis, V., Ioannidis, N., Karigiannis, J., Tsotros, M., Gounaris, M., Stricker, D., Gleue, T., Daehne, P., Almeida, L. (2002). Archeoguide: An augmented reality guide for archaeological sites. *IEEE Computer Graphics and Applications* **22**(5): 52–60
 194. Wallis, A. (2007). Is Modding Useful? In: *Game Career Guide 2007*, pp 25–28. CMP Media

195. Wand, M., Straßer, W. (2003). Real-time caustics. In: P. Brunet, D. Fellner (eds) *Computer Graphics Forum*, vol. 22, 3
196. Waring, P. (2007). Representation of ancient warfare in modern video games. Master's thesis, School of Arts, Histories and Cultures, University of Manchester (2007)
197. Watt, A., Policarpo, F. (2005). *Advanced Game Development with Programmable Graphics Hardware*. A. K. Peters
198. Wright, T., Madey, G.: A survey of collaborative virtual environment technologies. Tech. Rep. 2008-11, University of Notre Dame, USA (2008)
199. Wyman, C. (2007). Interactive refractions and caustics using image-space techniques. In: W. Engel (ed) *Shader X5: Advanced Rendering Techniques*, pp 359–371. Charles River Media
200. Yin, P., Jiang, X., Shi, J., Zhou, R. (2006). Multi-screen tiled displayed, parallel rendering system for a large terrain dataset. *IJVR* **5**(4): 47–54
201. Yu, J., Yang, J., McMillan, L. (2005). Real-time reflection mapping with parallax. In: *I3D '05: Proceedings of the 2005 symposium on Interactive 3D graphics and games*, pp 133–138
202. Zerbst, S., Düvel, O., Anderson, E. (2003). *3D-Spieleprogrammierung*. Markt + Technik
203. Zhou, T., Chen, J.X., Pullen, M. (2007). Accurate depth of field simulation in real time. *Computer Graphics Forum* **26**(1): 655–664
204. Zyda, M. (2005). From visual simulation to virtual reality to games. *IEEE Computer* **38**(9): 25–32